

Unit2-P: Techniques:

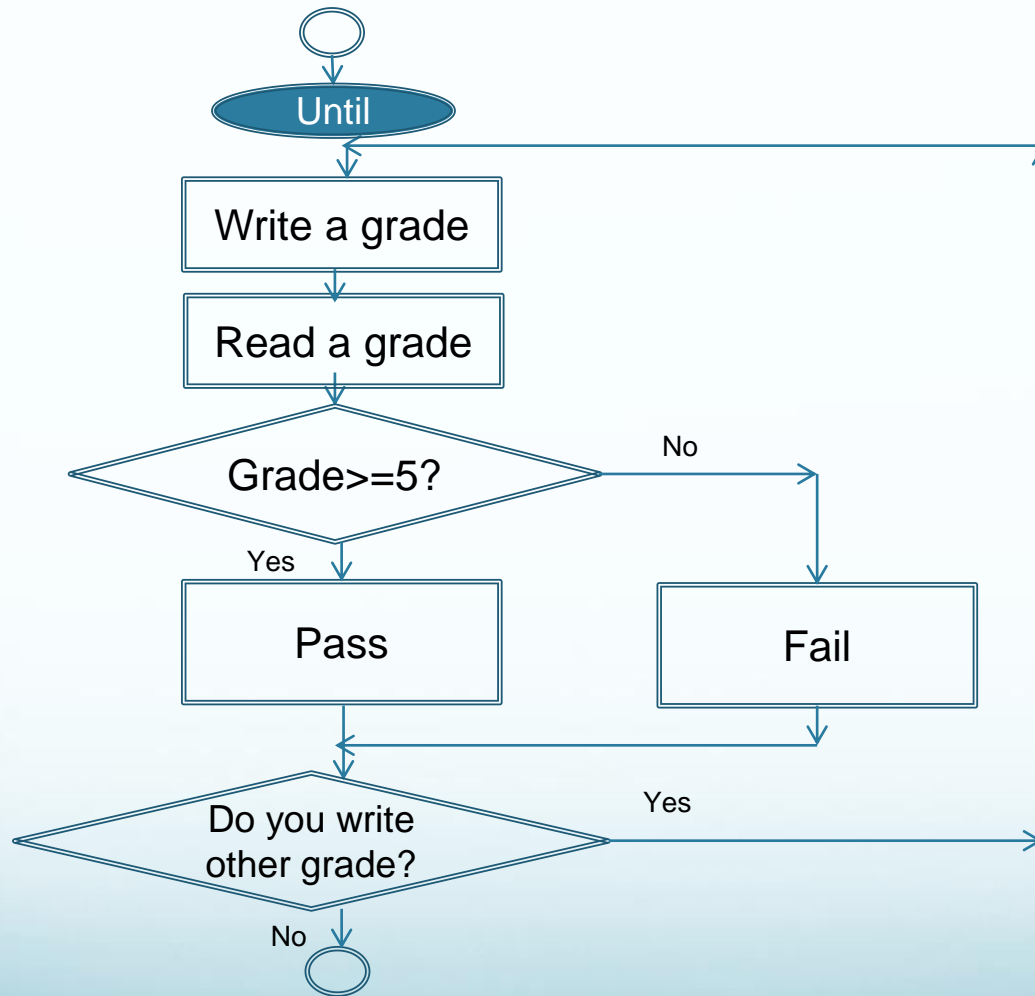
- Flowchart
- Pseudocode
- Trace



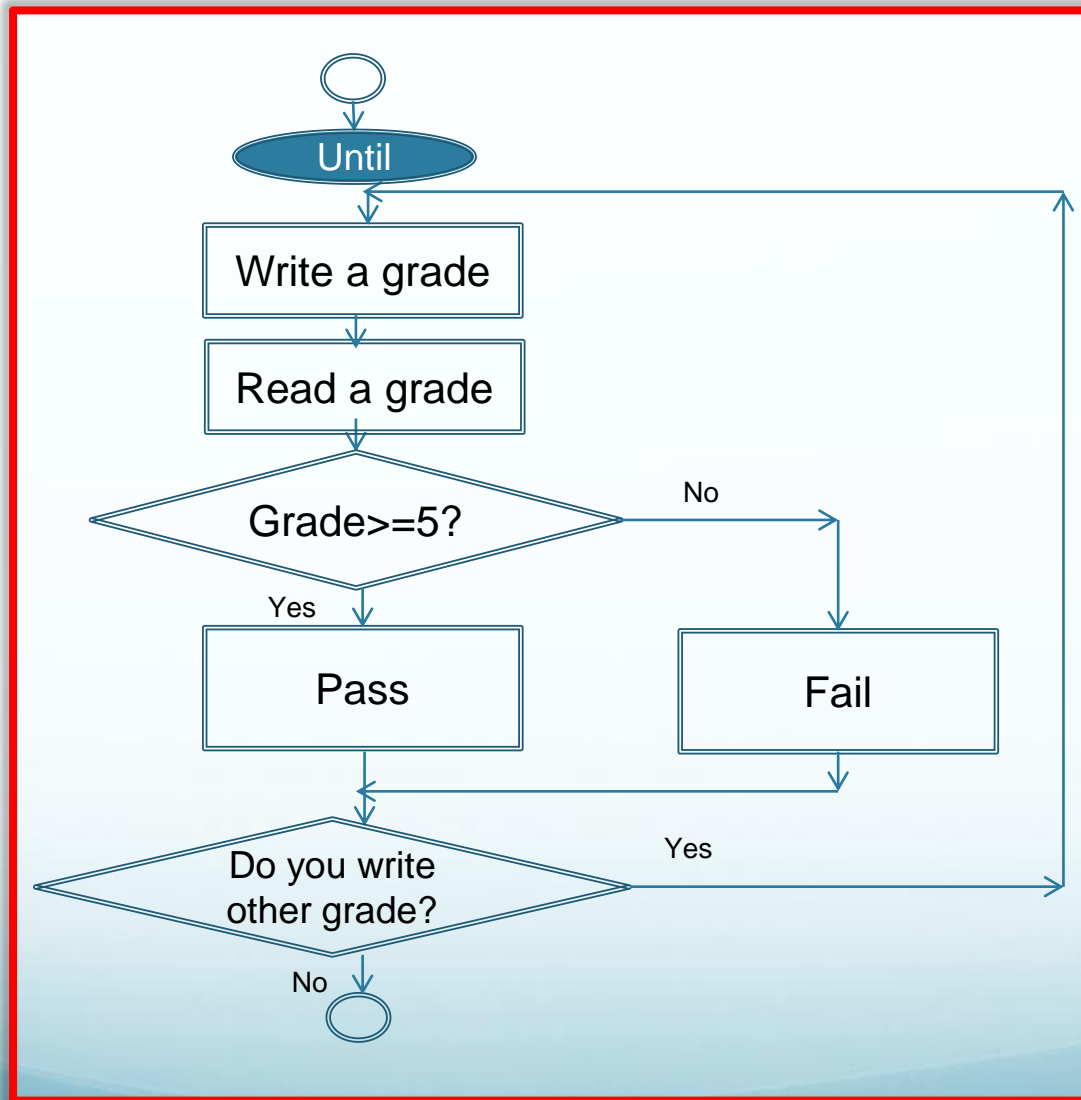
Techniques

- **Techniques:**
 - **Flowchart:** analyze programming functionality;
 - **Pseudocode:** convert the flowchart into a language understandable by any person who is unfamiliar with the R language;
 - **Trace:** validate and verify that the programmed code returns the desired result.

What does this program do?



What does this program do?



Flowchart

- **Instructions**

- **Sequence:** instruction is not executed until the previous one ends.
- **Selection:** instruction that allows the execution of the program to be bifurcated according to a established logical condition.
- **Iteration:** instruction that allows repeating the execution of the instructions while a condition is fulfilled

Flowchart: Symbolology

- Begin/End: 

- Sentence: 

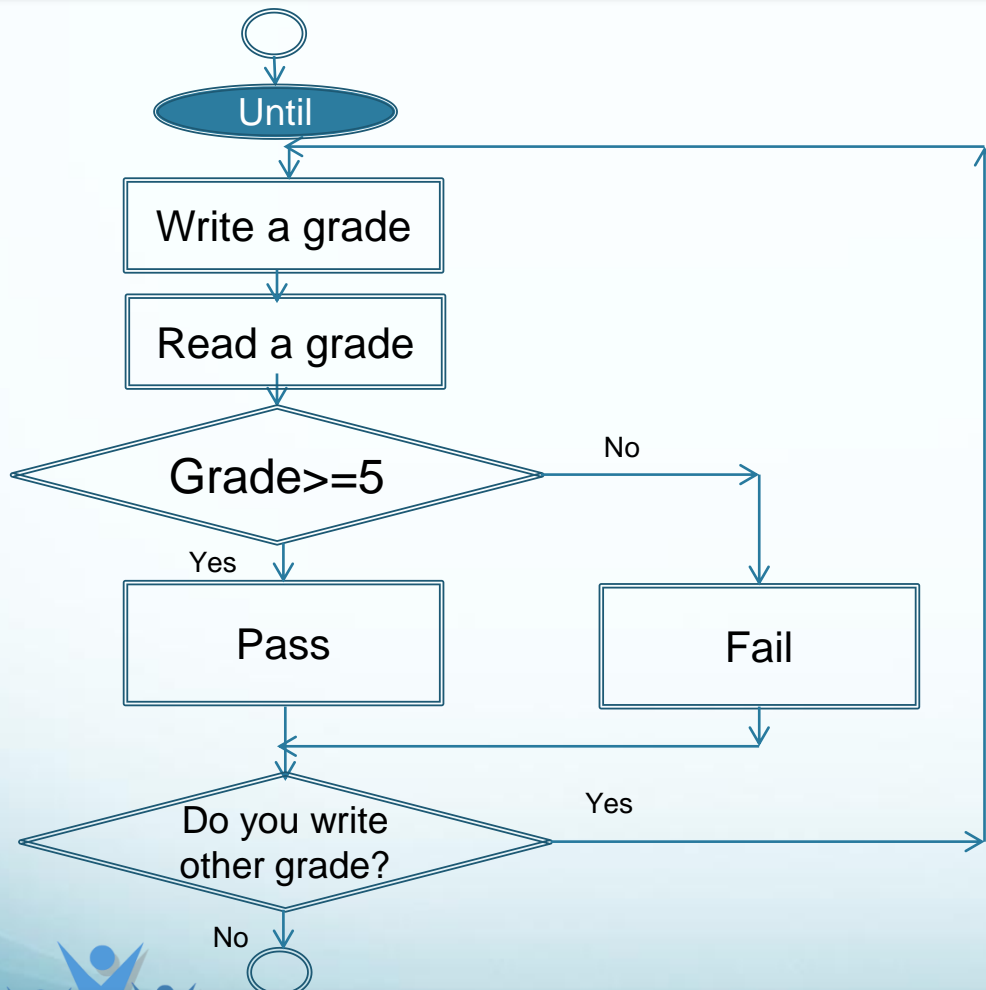
- Flowline: 

- Decision: 

Flowchart: Rules

- Flowcharts should be written from top to bottom, and/or from left to right.
- The symbols are joined with lines, which have at the tip an arrow that indicates the direction that flows the information processes, you should use only horizontal or vertical flow lines (never diagonal).
- Crossing of lines should be avoided.
- There should be no flow lines unconnected
- All text written within a symbol must be legible, precise, avoiding the use of many words.
- All symbols can have more than one entry line, except for the final symbol.

How do I translate this flowchart to code?



```
Start
repeat
  show "Write a grade:"
  read (grade)
  if (grade >= 5)
    show "Pass"
  if not
    show "Fail"
  show "Do you want to write other grade?"
  read (answer)
  if (answer = "No")
    exit
End
```


Pseudocode

- It is a language between the natural language of the programmer and the selected programming language.
- There is no standard syntax for the pseudocode
- Features:
 - It is easy to learn and use.
 - It is concise.
 - It is independent of the programming language that will be used.
 - It facilitates the passage of the program to the programming language.
 - It is easy to maintain.
 - Disadvantages: lack of standardization and difficulty reading when its size grows.

Pseudocode

- Writing:
 - It is read from up to down
 - It has a start and an end
 - In bifurcation and conditional structure: the block of instructions that is carried out when the condition is fulfilled, must be TABULATED TOWARDS THE RIGHT

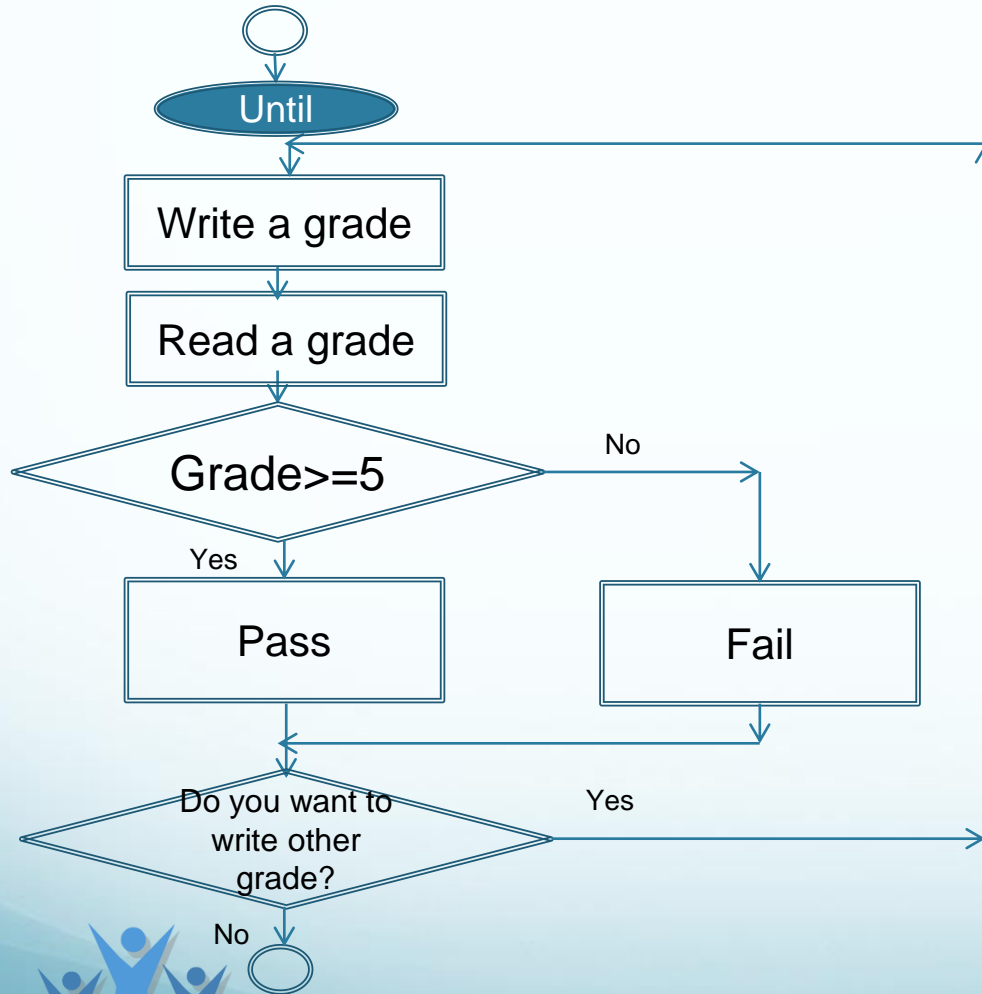
Pseudocode -> Code

```
Start
repeat
  show "Write a grade:"
  read (grade)
  if (grade>=5)
    show "Pass"
  if not
    show "Fail"
  show "Do you want to write other grade?"
  read (answer)
  if(answer="No")
    exit
End
```

```
grade<-function()
{
  repeat
    print("Write a grade:")
    grade=scan(,,1)
    if (grade>=5)
      print ("Pass")
    else
      print("Fail")
    print(«Do you want to wrte other grade:")
    answer=scan(,,1)
    if (answer=="No")
      break
}
```

How do I check that this program is right?

- Check if the students have passed or failed.



```
grade<-function()
{
  repeat
    print("Write a grade:")
    grade=scan(,,1)
    if (grade>=5)
      print ("Pass")
    else
      print("Fail")
      print("Do you want to write other grade:")
      answer=scan(,,1)
      if (answer=="No")
        break
  }
}
```

Grade	
5	Pass
4	Fail
8	Pass

Trace

- Process to go sentence to sentence and note variables value to check if the code implemented is correct
- Steps:
 - On a blank sheet, put a row with all the variables that appear in the code.
 - Each variable will have a column
 - Execute each sentence and if variables value changes in the sentence, write the new value in the variable column just below the previous value.